```
import pandas as pd
Data1 = pd.read_csv("Data1.csv")
Data1

seq = list(Data2['Series'])

def sim1(a,b):
  if a*b < 0:
    return 0
  return min(a,b)/max(a,b)

def dissim1(a,b):
  return 1-sim1(a,b)

def simhelper1num(seq):
  resnum = 0
  for i in range(len(seq)):
    if i == 0:
      resnum += seq[i] * sim1(seq[i],seq[i])
    else:
      for j in range(0,i+1):
        if (seq[j]<=seq[i]) :
          resnum += (seq[i] * sim1(seq[i],seq[j]))
        else :
          resnum += (seq[i] * -1 * sim1(seq[i],seq[j]))
  return resnum


def simhelper1den(seq):
  resden = 0
  for i in range(len(seq)):
      if i==0:
         resden += sim1(seq[i],seq[i])
      else:
        for j in range(0,i+1):
          resden += sim1(seq[i],seq[j])
  return resden


def fynplus1sim(seq):
  return (simhelper1num(seq) / simhelper1den(seq))


def dissimhelper1num(seq):
  resnum = 0
  for i in range(len(seq)):
    if i == 0:
      resnum += seq[i] * dissim1(seq[i],seq[i])
    else:
      for j in range(0,i+1):
        if ( seq[j]<=seq[i]):
          resnum += (seq[i] * dissim1(seq[i],seq[j]))
        else:
          resnum += (seq[i] * -1 * dissim1(seq[i],seq[j]))
  return resnum


def dissimhelper1den(seq):
     resden = 0
     for i in range(len(seq)):
       if i==0:
         resden += dissim1(seq[i],seq[i])
       else:
         for j in range(0,i+1):
           resden += dissim1(seq[i],seq[j])
     return resden


def fynplus1dissim(seq):
  return (dissimhelper1num(seq) / dissimhelper1den(seq))


def fynplus1(seq):
  return fynplus1dissim(seq)+fynplus1sim(seq)


seq = [1,2,3]
print(fynplus1(seq))


seq = [2,3,5,7,11,13,17,19]
print(fynplus1(seq))
```

```
seq = [3,2,7]
print(fynplus1(seq))

seq = [19,17,13,11,7,5,3,2]
print(fynplus1(seq))

seq = [-5.1451,2,3,5,7,11,13,17,19]
print(fynplus1(seq))

# all subseries

# using a function

def highlyused(seq):
  dupseq = seq.copy()
  dupseq.append(0)
  lst = []
  for j in range(2,len(dupseq)+1):
    temp=[]
    for i in range(len(dupseq)-1,-1,-j):
      temp.append(dupseq[i])
    lst.append(temp)

# accessing only subseries with atleast 2 elements excluding 0
  for i in range(len(lst)):
```

```
  if len(lst[i])<=2:
   inter = lst[:i]
   break

# removing 0

 for i in inter:
  i.pop(0)

# reversing the subseries

 for i in range(len(inter)):
  inter[i].reverse()

# inserting the origial seq at the start
 inter.insert(0,seq)
 inter

 return inter

seq = list(Data2['Series'])

def wgtavg(seq):
 fin = highlyused(seq)
 print("subseries : ",fin)
```

```
  sscosfa = []
  for i in fin:
    sscosfa.append(fynplus1(i))
  # print("subseriescosfa : ",sscosfa)

  weights = []
  for i in sscosfa:
    weights.append(i/sum(sscosfa))
  # print("weights of subseries : ",weights)

  weightedaverage = 0
  comb = list(zip(sscosfa,weights))
  for i in comb:
    weightedaverage += (i[0]*i[1])
  # print("Weighted average : ",weightedaverage)
  return weightedaverage

def allsswgtavgxast(fin):
  wts = []
  for i in fin:
    wts.append(wgtavg(i))
  return wts

SSweights = allsswgtavgxast(fin)
```

```
SSweights


def finalwtdavg(SSweights):
  finalres = 0
  for i in SSweights:
    finalres += (i/sum(SSweights))*i
  return finalres


print(finalwtdavg(SSweights))
```